Quantum tangent kernel

Norihito Shirai,[1,*] Kenji Kubo,[1,2,†] Kosuke Mitarai,[1,3,4,‡] and Keisuke Fujii[1,3,5,§]
[1]*Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan*
[2]*R4D, Mercari Inc., Roppongi Hills Mori Tower 18F, 6-10-1 Roppongi, Minato-ku, Tokyo 106-6118, Japan*
[3]*Center for Quantum Information and Quantum Biology, Osaka University, 1-2 Machikameyama, Toyonaka, Osaka 560-0043, Japan*
[4]*JST, PRESTO, 4-1-8 Honcho, Kawaguchi, Saitama 332-0012, Japan*
[5]*RIKEN Center for Quantum Computing, Wako, Saitama 351-0198, Japan*

(Received 24 November 2022; accepted 3 July 2024; published 16 August 2024)

The quantum kernel method is one of the key approaches to quantum machine learning, which has the advantage of not requiring optimization and its theoretical simplicity. By virtue of these properties, several experimental demonstrations and discussions of the potential advantages have been developed so far. However, as is the case in classical machine learning, not all quantum machine learning models could be regarded as kernel methods. In this work, we explore a quantum machine learning model with a deep parametrized quantum circuit and aim to go beyond the conventional quantum kernel method. In this case, the expressive power and performance are expected to be enhanced, while the training process might be a bottleneck because of the barren plateaus. Moreover, the high computational cost of gradient-based optimization and the large search space of the gradient-free optimization directly make the training intractable. However, we find that parameters of a deep enough quantum circuit do not move much from their initial values during training, allowing for a first-order expansion with respect to the parameters. This behavior is similar to that of the neural tangent kernel in classical machine learning, and such a quantum machine learning with deep variational quantum circuits can be described by another emergent kernel, the *quantum tangent kernel*. We show that the proposed quantum tangent kernel has the potential to outperform the conventional quantum kernel method by performing a classification task on an ansatz-generated dataset. This work provides a different direction beyond the conventional quantum kernel method and explores the potential power of quantum machine learning with deep parametrized quantum circuits.

DOI: 10.1103/PhysRevResearch.6.033179

I. INTRODUCTION

Applying quantum computers to machine learning purposes is an emerging area of research. In particular, motivated by the recent advance of hardware technology [1], techniques to apply so-called noisy intermediate-scale quantum (NISQ) devices [2] have rapidly developed [3–7]. One direction that is frequently explored is variational methods, which use parametrized quantum circuits to construct a model $y(\boldsymbol{x}, \boldsymbol{\theta})$. Such a model outputs a prediction when fed with the input data $\boldsymbol{x}$. More specifically, using a parametrized quantum circuit $U(\boldsymbol{x}, \boldsymbol{\theta})$ with trainable parameters $\boldsymbol{\theta}$, we construct a model $y(\boldsymbol{x}, \boldsymbol{\theta})$ for an input $\boldsymbol{x}$ by the expectation value of an observable O: $y(\boldsymbol{x}, \boldsymbol{\theta}) = \langle 0| U^\dagger(\boldsymbol{x}, \boldsymbol{\theta}) O U(\boldsymbol{x}, \boldsymbol{\theta}) |0\rangle$. Since there are quantum circuits that are hard to simulate classically, we might be able to construct a machine learning model that exceeds the capability of classical computers.

Another promising direction is the so-called quantum kernel method [4,5,8]. In this approach, we use quantum computers for preparing a classically intractable feature vector $|\phi(\boldsymbol{x})\rangle$ and taking the inner products of the feature vectors. The training is performed on a classical computer using the values of the inner product between each pair of training data points. Since training the quantum circuit is often difficult in the aforementioned variational methods, quantum kernel methods are advantageous in that we do not have to optimize the quantum circuit. Moreover, it can be shown that the quantum kernel method outperforms the variational ones in a certain sense [8]; if $U(\boldsymbol{x}, \boldsymbol{\theta})$ takes the form of $U(\boldsymbol{x}, \boldsymbol{\theta}) = V(\boldsymbol{\theta}) U_\phi(\boldsymbol{x})$, then the quantum kernel method with feature vector $|\phi(\boldsymbol{x})\rangle = U_\phi(\boldsymbol{x}) |0\rangle$ performs at least as well as the variational method on the training dataset. Its simple framework greatly advanced the construction of quantum machine learning theory, providing insights on potential advantages [9,10]. Also, owing to its experimental easiness, there have been several experimental demonstrations of the method [11–13]. However, as is well known in the machine learning field, not all machine learning models can be described by kernel methods, and there must be approaches that go beyond them, such as deep learning.

In this work, we explore how to construct a quantum machine learning model that goes beyond the conventional

*Contact author: shirai@qc.ee.es.osaka-u.ac.jp
†Contact author: kenji.kubo.q@gmail.com
‡Contact author: mitarai@qc.ee.es.osaka-u.ac.jp
§Contact author: fujii@qc.ee.es.osaka-u.ac.jp

quantum kernel method. To this end, we investigate the performance of a model using deep quantum circuits in the form of $U(\boldsymbol{x},\boldsymbol{\theta}) = \prod_{i=0}^{L} V_i(\boldsymbol{\theta}_i)U_{\phi,i}(\boldsymbol{x})$ $(L > 0)$. The circuit is now not in the form of $U(\boldsymbol{x},\boldsymbol{\theta}) = V(\boldsymbol{\theta})U_\phi(\boldsymbol{x})$, and hence this model cannot be directly translated to a kernel framework. If we add ancilla qubits to the original quantum circuit, the above model can approximate or exactly represent the kernel framework [14]. However, in the NISQ era, increasing the number of qubits is not the best strategy.

Unfortunately, as parametrized quantum circuits become deeper and go beyond the conventional quantum kernel method, it becomes more difficult to train the parameters as was the case with deep neural networks in the classical literature. While this issue would be resolved by finding a good ansatz and/or initial parameters [15–17], we consider an alternative approach, i.e., overparametrization [18–20].

In this work, we find that when the circuit is deep enough, each of the parameters in $\boldsymbol{\theta}$ does not move much and stays close to the initial random guess during training with gradient descent. This behavior is similar to classical deep neural networks [21], where the amount of change in the parameters is small, and the network is well described by a linear model with the tangent on the parameters as the basis functions. Thus optimized parameters on overparametrized networks can be found by another kernel, the so-called neural tangent kernel [21]. The observation motivates us to propose the quantum tangent kernel (QTK), which is a quantum analog of the neural tangent kernel.

Specifically, we define the QTK by the kernel associated with the feature map $\boldsymbol{x} \to \nabla_{\boldsymbol{\theta}} y(\boldsymbol{x},\boldsymbol{\theta})$ for a deep parametrized quantum circuit. The QTK can be calculated efficiently on a quantum computer by using analytical differentiation of parametrized quantum circuits, the so-called parameter-shift rule [3,22]. Then, the QTK combined with the standard kernel methods such as the support vector machine (SVM) allows us to learn and infer without any explicit optimization of the parametrized quantum circuit. We compare the performance of the QTK based on the deep parametrized quantum circuit and the conventional quantum kernel method for an "ansatz-generated" dataset generated by a deep parametrized quantum circuit, while the parameters are randomly chosen apart from those in the QTK. As a result, we find that the QTK outperforms the conventional quantum kernel method. This implies that deep parametrized quantum circuits have a great potential. This work opens up a different direction beyond the conventional quantum kernel method for deeper quantum machine learning.

II. BACKGROUND

A. Kernel methods

In this subsection, we explain the kernel method and the SVM, which is one of the kernel-based classification methods. Throughout this paper, we denote the training dataset by $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, where $\boldsymbol{x}_i$ is the input data and y_i is the corresponding teacher data.

The kernel method is a technique for mapping features to a higher-dimensional feature space in order to introduce nonlinearity to the model. It employs a nonlinear map ϕ from the original space to the higher-dimensional feature space:

$$\phi : \chi \to \mathcal{H}, \quad \boldsymbol{x}_i \to \phi(\boldsymbol{x}_i), \tag{1}$$

where χ is the original space of the data and $\mathcal{H}$ is a higher-dimensional feature space. In the kernel method, we only use inner products of ϕ between different input data. For two data $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, we define

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle, \tag{2}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product on $\mathcal{H}$. The kernel method has the advantage that it does not require direct computation of the vectors in high-dimensional feature space as long as their inner product can be calculated efficiently. This inner product represents the similarity between the features and is called the kernel function.

As an example of the kernel method, we consider the SVM. In the SVM, optimization of a linear model can be formulated as a quadratic programming problem. This primal problem is equivalent to its dual optimization problem. Therefore, the SVM with kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is trained by maximizing the following objective function with respect to the dual variable α_i:

$$L_D(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=1}^{N} \alpha_i, \tag{3}$$

subject to the constraints $\sum_{i=0}^{N} \alpha_i y_i = 0$ and $0 \leqslant \alpha_i \leqslant C$, where C is the regularization parameter. $L_D(\boldsymbol{\alpha})$ depends only on the kernel function, and hence we do not need to explicitly calculate the feature map ϕ. The solution α_i^* to the above optimization problem is used for building the prediction model as follows:

$$y(\boldsymbol{x}) = \mathrm{sign}\left(\sum_{i=1}^{N} y_i \alpha_i^* K(\boldsymbol{x}_i, \boldsymbol{x}) + b \right), \tag{4}$$

where the bias b is calculated as

$$b = y_j - \sum_{i=1}^{N} \alpha_i^* y_i K(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{5}$$

Although the formula of bias b holds for any j, practically, the average for all j's is taken. Kernel methods are not limited to classical machine learning. The quantum kernel can be calculated by taking the inner product of quantum states, and approaches using it are known as quantum kernel methods.

B. Neural tangent kernel

Let us denote by $f(\boldsymbol{x}_i, \boldsymbol{\theta})$ the output of a neural network where $\boldsymbol{\theta}$'s are parameters in the network, and $\boldsymbol{x}_i$ is the input data. $f(\boldsymbol{x}_i, \boldsymbol{\theta})$ is a nonlinear function with respect to $\boldsymbol{x}, \boldsymbol{\theta}$ and is usually optimized by gradient descent. In the large-width overparametrized neural network, the values of the parameters change only slightly from the initial values during training even if the initial values of parameters are set randomly [21]. Therefore, the output of such a neural network can be approximated well by the first-order expansion with respect to the parameters around the initial values:

$$f(\boldsymbol{x}_i, \boldsymbol{\theta}) \simeq f(\boldsymbol{x}_i, \boldsymbol{\theta}_0) + \nabla_\theta f(\boldsymbol{x}_i, \boldsymbol{\theta}_0)^T (\boldsymbol{\theta} - \boldsymbol{\theta}_0), \tag{6}$$

where $\boldsymbol{\theta}_0$'s are the initial values of parameters of a neural network. This approximation allows us to interpret the neural network as a linear model for $\boldsymbol{\theta}$, with the feature map

$$\phi(\boldsymbol{x}) = \nabla_\theta f(\boldsymbol{x}, \boldsymbol{\theta}_0)\,. \tag{7}$$

Using this feature map, we define the following kernel called the neural tangent kernel (NTK) [21,23]:

$$K_{\rm ntk}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \nabla_\theta f(\boldsymbol{x}_i, \boldsymbol{\theta}_0)^T \nabla_\theta f(\boldsymbol{x}_j, \boldsymbol{\theta}_0)\,. \tag{8}$$

By their construction, NTK-based linear models are expected to be equivalent to the large-width neural network as long as the assumption Eq. (6) is valid.

C. Quantum machine learning as a kernel method

Conventional variational quantum machine learning models [3–5] work in the following manner. First, data are encoded into quantum states by $U_\phi(\boldsymbol{x})$. Then, we apply a trainable parametrized circuit $V(\boldsymbol{\theta})$. Finally, we measure the expectation value of an observable O, which is used as the model output $y(\boldsymbol{x}, \boldsymbol{\theta})$. Mathematically, the above process can be written as

$$y(\boldsymbol{x}, \boldsymbol{\theta}) = \langle 0^n | U_\phi^\dagger(\boldsymbol{x}) V^\dagger(\boldsymbol{\theta}) O V(\boldsymbol{\theta}) U_\phi(\boldsymbol{x}) | 0^n \rangle, \tag{9}$$

where $|0^n\rangle$ denotes an n-qubit computational basis state. The linearity of this model can be readily seen by rewriting the above expression as

$$y(\boldsymbol{x}, \boldsymbol{\theta}) = {\rm Tr}[O(\boldsymbol{\theta})\rho(\boldsymbol{x})], \tag{10}$$

where

$$O(\boldsymbol{\theta}) = V^\dagger(\boldsymbol{\theta}) O V(\boldsymbol{\theta}), \tag{11}$$

$$\rho(\boldsymbol{x}) = U_\phi(\boldsymbol{x}) |0^n\rangle \langle 0^n| U_\phi^\dagger(\boldsymbol{x}). \tag{12}$$

Since ${\rm Tr}(A^\dagger B)$ for operators A and B defines an inner product in the operator space, Eq. (10) defines a linear model using the feature vector $\rho(\boldsymbol{x})$ and the weight vector $O(\boldsymbol{\theta})$ [8]. Therefore, if we construct a kernel-based model using the same feature vector $\rho(\boldsymbol{x})$, it performs at least as well as the model in Eq. (10) on a training dataset [8]. In this case, we define the kernel function as

$$K_q(\boldsymbol{x}_i, \boldsymbol{x}_j) = {\rm Tr}[\rho(\boldsymbol{x}_i)\rho(\boldsymbol{x}_j)] \tag{13}$$

$$= |\langle \phi(\boldsymbol{x}_i) | \phi(\boldsymbol{x}_j) \rangle|^2, \tag{14}$$

where $|\phi(\boldsymbol{x})\rangle$ is defined as follows:

$$|\phi(\boldsymbol{x})\rangle = U_\phi(\boldsymbol{x}) |0^n\rangle. \tag{15}$$

We call the kernel methods that are based on $K_q(\boldsymbol{x}_i, \boldsymbol{x}_j)$ the conventional quantum kernel method. The above argument only holds for a quantum model $y(\boldsymbol{x}, \boldsymbol{\theta}) = \langle 0^n | U^\dagger(\boldsymbol{x}, \boldsymbol{\theta}) O U(\boldsymbol{x}, \boldsymbol{\theta}) | 0^n \rangle$, with $U(\boldsymbol{x}, \boldsymbol{\theta})$ in the form of $V(\boldsymbol{\theta})U_\phi(\boldsymbol{x})$. If we can add auxiliary qubits, models not of the form $U(\boldsymbol{x}, \boldsymbol{\theta}) = V(\boldsymbol{\theta})U_\phi(\boldsymbol{x})$ can be approximated by a kernel method [14]. However, in the NISQ era, the approach of increasing the number of qubits is not feasible. In Sec. IV, we propose quantum circuits that cannot be split in such a way. This modification prevents us from directly rewriting the model into the form of Eq. (10), and thus we cannot construct an equivalent kernel model without introducing ancilla qubits as in Ref. [14].

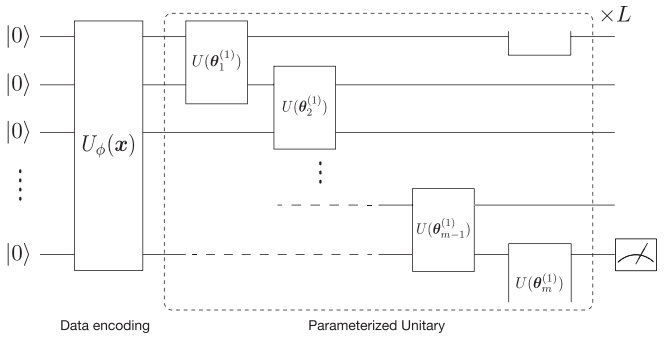

FIG. 1. The m-qubit ansatz used for numerical simulations. $U_\phi(\boldsymbol{x})$ is the quantum feature map for encoding classical data. $U(\boldsymbol{\theta}_j^{(i)}) \in SU(4)$ is the parametrized unitary of the ith layer. At the output, we measure the Pauli Z expectation value of the final qubit.

III. QUANTUM TANGENT KERNEL

In this section, we apply the formulation of the NTK described in Sec. II B to parametrized quantum circuits. We consider a model whose output is given as the expectation value of an operator O as

$$y(\boldsymbol{x}, \boldsymbol{\theta}) = \langle 0^n | U^\dagger(\boldsymbol{x}, \boldsymbol{\theta}) O U(\boldsymbol{x}, \boldsymbol{\theta}) | 0^n \rangle, \tag{16}$$

where $\boldsymbol{x}$ is the input data and $\boldsymbol{\theta}$ are parameters of a quantum circuit. We define the following kernel by using the output of a quantum circuit analogous to the NTK:

$$K_{\rm qtk}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \nabla_\theta y(\boldsymbol{x}_i, \boldsymbol{\theta}_0)^T \nabla_\theta y(\boldsymbol{x}_j, \boldsymbol{\theta}_0). \tag{17}$$

We call the kernel $K_{\rm qtk}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ the QTK. As in NTK, the QTK approximates the original model $y(\boldsymbol{x}, \boldsymbol{\theta})$ well as long as the parameters do not change much from their initial random guess $\boldsymbol{\theta}_0$ when, for example, updating them by gradient descent based on some suitable cost function.

We can numerically check that such a phenomenon occurs for quantum circuits with a large number of layers. To this end, we train quantum circuits with varying numbers of layers and look at the changes of their parameters.

Here, we consider two types of quantum tangent kernel according to how the data are encoded into quantum states. First one is a circuit where $\boldsymbol{x}$ is encoded only at the first layer as in Fig. 1:

$$U_{\rm shallow}(\boldsymbol{x}, \boldsymbol{\theta}) = V(\boldsymbol{\theta}) U_\phi(\boldsymbol{x}), \tag{18}$$

where $V(\boldsymbol{\theta})$ is a parametrized unitary and $U_\phi(\boldsymbol{x})$ is a quantum feature map to encode data. This type of quantum circuit can be interpreted as the conventional quantum kernel method. We call the QTK associated with this type of ansatz a *shallow QTK*. Next, in order to increase the nonlinearity with respect to $\boldsymbol{x}$, we consider a multilayered circuit that alternates between data encoding and a parametrized unitary as shown in Fig. 2. More concretely, we consider the following unitary:

$$U_{\rm deep}(\boldsymbol{x}, \boldsymbol{\theta}) = \prod_{i=1}^{L} [V(\boldsymbol{\theta}_i) U_\phi(\boldsymbol{x})] \quad (L > 1). \tag{19}$$

We call the QTK associated with this type of ansatz a *deep QTK*. The same type of ansatz is proposed in Ref. [14] and is called a data re-uploading circuit (see *Note added*). The deep

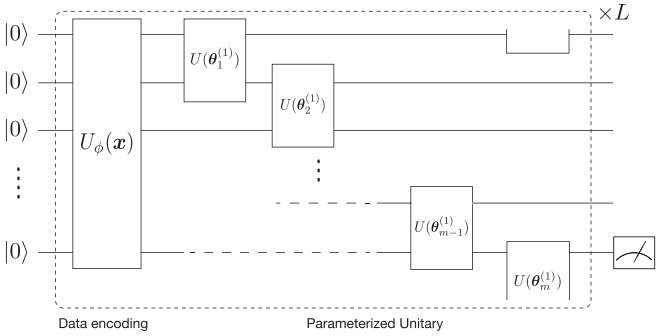

FIG. 2. $U_\phi(\boldsymbol{x})$ is the feature map and $U(\boldsymbol{\theta}_j^{(i)}) \in SU(4)$ is the parametrized unitary of the ith layer. To increase the nonlinearity of the kernel, the feature map $U_\phi(\boldsymbol{x})$ and the parametrized unitary are iteratively applied. At the output, we measure the Pauli Z expectation value of the final qubit.

QTK contains higher-order terms about data $\boldsymbol{x}$ and has higher nonlinearity with respect to $\boldsymbol{x}$. We demonstrate the expressive power of this type of QTK in Sec. IV.

We train those two types of ten-qubit quantum circuits as shown in Figs. 1 and 2 on the MNIST dataset [24] reduced to a ten-dimensional dataset by principal component analysis and use only the digits 8 and 3 for binary classification. In this numerical experiment, the quantum feature map to encode the data is given by $U_\phi(\boldsymbol{x}) = \bigotimes_{i=1}^{10} \exp(ix_iY_i)$, where Y_i is the Pauli Y operator acting on the ith qubit. We apply $U(\boldsymbol{\theta}_i^{(j)}) \in SU(4)$ on each neighboring qubit as in Figs. 1 and 2. SU(4) is parametrized according to Cartan decomposition as follows:

$$U\left(\boldsymbol{\theta}_i^{(j)}\right) = k_1 \exp\left[\frac{i}{2}(\theta_{xx}X_iX_{i+1} + \theta_{yy}Y_iY_{i+1} + \theta_{zz}Z_iZ_{i+1})\right]k_2, \tag{20}$$

where $k_1, k_2 \in \mathrm{SU}(2) \otimes \mathrm{SU}(2)$ and X_i, Y_i, and Z_i are the Pauli operators.

The initial values of the parameters of these quantum circuits are set randomly using a uniform distribution between 0 and 2π. We use the mean squared error (MSE) as the loss function and the standard stochastic gradient descent with minibatch size 64 to train the quantum circuits. The expectation value of the deep quantum circuit concentrates on its average over Hilbert space due to barren plateaus [25]. In our case, the expectation value of the observable $O = Z_{10}$ on the deep quantum circuit concentrates on 0. Now we use the MSE as the loss function and the target value is 1 or -1 in the classification problem. We classify a sample $\boldsymbol{x}_i$ into class 1 if $y(\boldsymbol{x}_i) \geqslant 0$ and into class -1 if $y(\boldsymbol{x}_i) \leqslant 0$. Even if the classification is correct, if the expectation value is close to 0, the MSE will not decrease since the squared error between the target and the prediction value $[y_i - y(\boldsymbol{x}_i)]^2$ is large. For this reason, we multiply a scale factor to the expectation value to verify that the training is successful, i.e., $y(\boldsymbol{x}, \boldsymbol{\theta}) = C\langle 0^n|U^\dagger(\boldsymbol{x}, \boldsymbol{\theta})Z_{10}U(\boldsymbol{x}, \boldsymbol{\theta})|0^n\rangle$. The variance of the expectation value of the quantum circuit multiplied by a scale factor should be roughly 1.0 to decrease the MSE. We chose $C = 4.0$ for a quantum circuit in Fig. 1 and $C = 10.0$ for the one in Fig. 2 to decrease the MSE. Figures 3(a) and 4(a) show the relative norm changes in the parameters of the

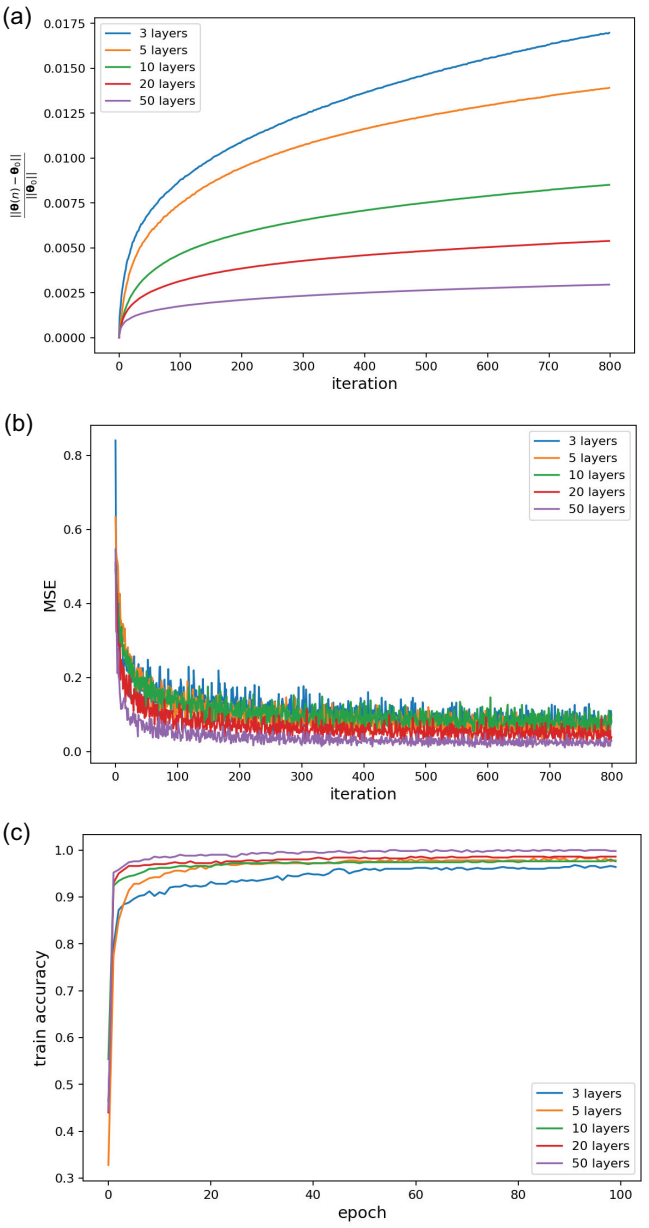

FIG. 3. (a) Relative norm change in the parameters of the quantum circuit in Fig. 1 from initial values during training by gradient descent. $\boldsymbol{\theta}(n)$ is the parameter at the nth iteration. $\boldsymbol{\theta}_0$ is the initial value of the parameter. Panels (b) and (c) show the behavior of training losses and training accuracies for different numbers of layers during training.

quantum circuits with layers $L = 3$, 5, 10, 20, and 50. We can observe that the changes of the parameters during training become small when a quantum circuit has more layers. Panels (b) and (c) in Figs. 3 and 4, which respectively show the decrease of MSE and the increase of training accuracy, indicate the success of training. In addition, we also show the distribution of the difference of the parameters from their initial values after training in Fig. 5. This result implies that it is possible to linearly approximate $y(\boldsymbol{x}, \boldsymbol{\theta})$ with respect to its parameters when the circuit is sufficiently deep. Hence, we

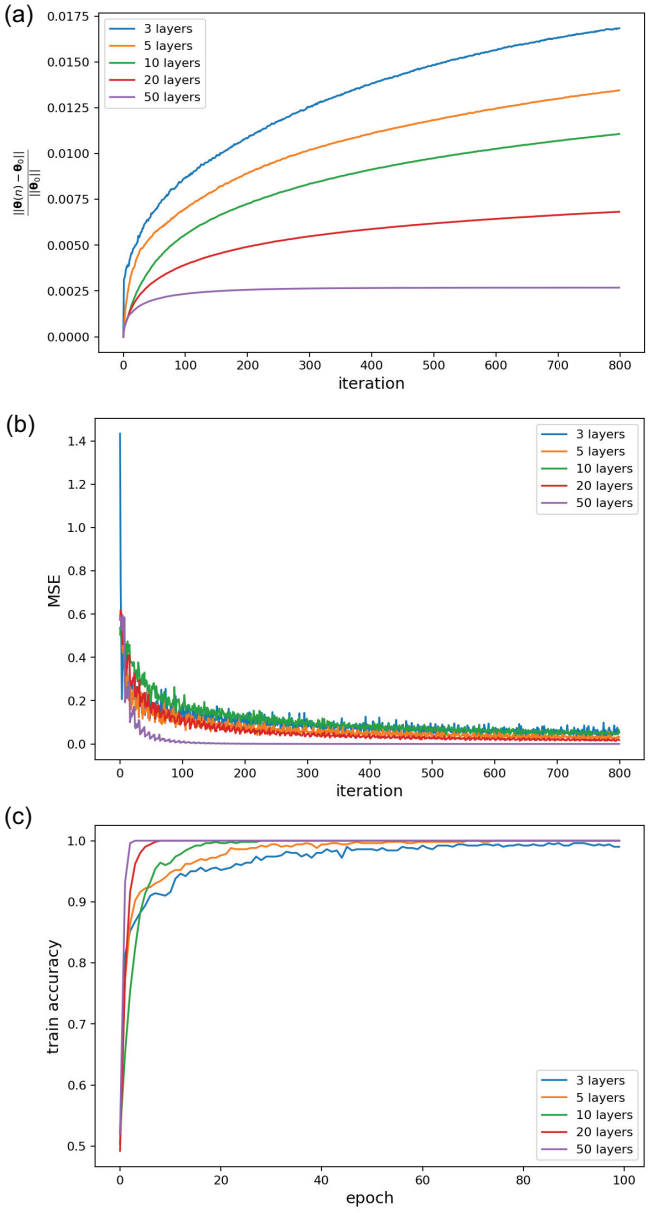

FIG. 4. The same numerical experiments as in Fig. 3 are performed for the quantum circuit in Fig. 2. Panels (a), (b), and (c) show the relative norm changes in the parameters, MSEs, and train accuracies for different numbers of layers during training.

can expect the QTK to provide a machine learning model that is approximately equivalent to $y(\boldsymbol{x}, \boldsymbol{\theta})$.

For a given ansatz, the QTK can be calculated on a quantum computer with parameter-shift rules [3,22] and their generalizations [26,27]. This provides us an alternative quantum machine learning model other than the conventional variational methods and quantum kernel methods. Note that, as opposed to the case of the NTK [21,28], currently we do not know how to calculate the QTK analytically for a given form of ansatz in the infinite depth limit. We leave this direction as an interesting future direction to explore. On the other hand, in the limit of a large number of qubits, an analytic solution that describes the convergence of the training error has been found [29].

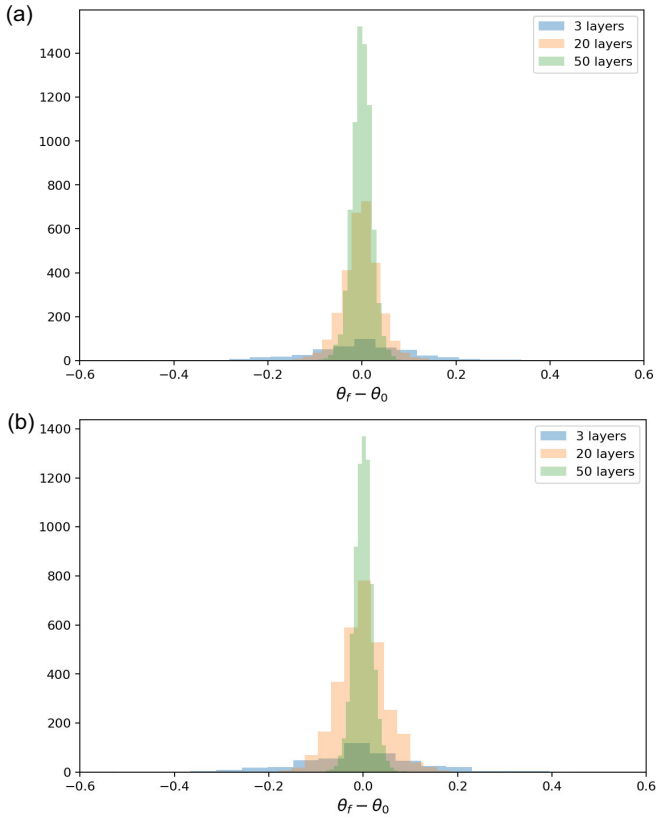

FIG. 5. The distribution of parameter changes from their initial values after quantum circuits are trained. θ_f is the value of parameter after training and θ_0 is its initial value. Panels (a) and (b) are for the quantum circuits in Figs. 1 and 2, respectively.

IV. EXPRESSIVE POWER OF QTK

We numerically demonstrate the expressive power of quantum tangent kernels. In order to demonstrate the expressive power, we generate an "ansatz-generated" dataset by using a quantum circuit for U_{deep} and classify the data using the SVM with the three types of kernels: shallow QTK, deep QTK, and conventional quantum kernel defined by the feature map $U_\phi(\boldsymbol{x})$. If these data are classified efficiently by the SVM with the deep QTK, it has higher expressive power than the other two.

In this numerical experiment, we use $U_{\boldsymbol{\phi}}(\boldsymbol{x})$ in the forms of

$$U_\phi(x_i) = \exp\left[i\phi(x_i)Y_i\right] \tag{21}$$

and

$$U_\phi(x_i, x_j) = \exp[i\phi_i(x_i, x_j)Y_i \otimes Y_j], \tag{22}$$

where Y_i is the Pauli Y operator acting on the ith qubit. We apply Eq. (21) to all qubits and Eq. (22) between neighboring qubits. The functions $\phi_i(\boldsymbol{x})$ and $\phi_{ij}(\boldsymbol{x})$ are given by

$$\phi_i(\boldsymbol{x}) = \arcsin(x_i), \tag{23}$$

$$\phi_{ij}(\boldsymbol{x}) = \arcsin(x_i x_j). \tag{24}$$

The "ansatz-generated" dataset is generated in the following manner. Four-dimensional random value data $\{\boldsymbol{x}_i\}$ are input into $U_{\text{deep}}(\boldsymbol{x}, \boldsymbol{\theta})$ consisting of $n = 4$ qubits and $L = 10$

TABLE I. Classification accuracy for SVMs with three types of kernels. Three SVMs classify the ansatz-generated dataset generated by a quantum circuit for the deep QTK as shown in Fig. 2.

Kernel	Accuracy
Quantum kernel	0.7842
Shallow quantum tangent kernel	0.7484
Deep quantum tangent kernel	0.812

layers. Then, we evaluate the expectation value

$$l(\boldsymbol{x}_i, \boldsymbol{\theta}) = \langle 0^n | U^\dagger_{\text{deep}}(\boldsymbol{x}_i, \boldsymbol{\theta}) Z_4 U_{\text{deep}}(\boldsymbol{x}_i, \boldsymbol{\theta}) | 0^n \rangle \tag{25}$$

with a randomly chosen $\boldsymbol{\theta}$. For each i, we label $y_i = 1$ if $l(\boldsymbol{x}, \boldsymbol{\theta}) \geqslant 0$ and $y_i = -1$ if $l(\boldsymbol{x}, \boldsymbol{\theta}) < 0$. We generate the 15 000 samples of the four-dimensional input data $\boldsymbol{x}_i$ and its label y_i. The values of the labels are roughly balanced between 1 and -1. They are randomly split into 10 000 training and 5000 test data.

We classify the above dataset with the SVM using three different kernels: the shallow QTK [Eq. (17) with $U(\boldsymbol{x}, \boldsymbol{\theta}) = U_{\text{shallow}}(\boldsymbol{x}, \boldsymbol{\theta})$ and $L = 10$ layers], the deep QTK [Eq. (17) with $U(\boldsymbol{x}, \boldsymbol{\theta}) = U_{\text{deep}}(\boldsymbol{x}, \boldsymbol{\theta})$ and $L = 10$ layers], and the conventional quantum kernel [Eq. (13)]. In order to calculate the QTK and the deep QTK, we randomly set parameter values of a quantum circuit using a uniform distribution between 0 and 2π. Note that the parameters used in this learning phase are different from the ones used for data generation. We calculated QTKs by the parameter-shift rule [3,22]. The regularization strength C in the SVM optimization in Eq. (3) is determined via cross-validation for each kernel. We split the training data into five parts and use one of them as the validation data.

The results of the classification task are listed in Table I. Among the three kernels, the deep QTK outperforms the other kernels. This result indicates that the deep QTK employs a feature map that cannot be expressed very accurately with other kernels. In contrast, the shallow QTK is essentially just a quantum kernel method using the feature map in Eq. (18), so its performance is not improved compared to that of the conventional quantum kernel as expected.

The expressive power of kernels can be illustrated more directly by visualizing the feature map of each kernel. Figure 6 shows the distribution of the ansatz-generated dataset generated by quantum circuits for $U_{\text{shallow}}(\boldsymbol{x}, \boldsymbol{\theta})$ and $U_{\text{deep}}(\boldsymbol{x}, \boldsymbol{\theta})$. In order to visualize the distribution, we generate two-dimensional data using the two-qubit quantum circuits. These distributions qualitatively show that $U_{\text{deep}}(\boldsymbol{x}, \boldsymbol{\theta})$ has a more complex structure and can express higher nonlinearity.

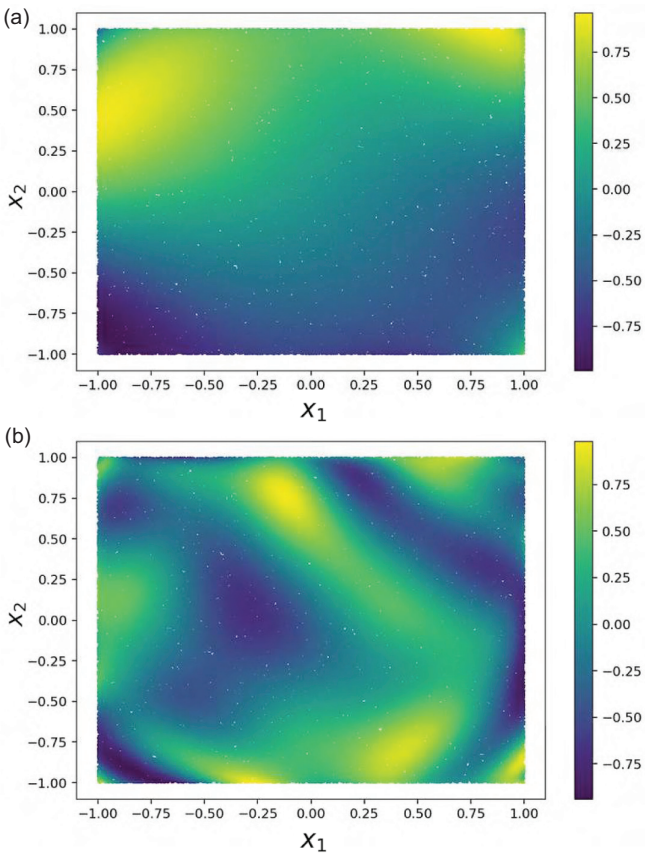

FIG. 6. (a) The distribution of outputs of the quantum circuit with $L = 10$ layers (Fig. 1) which can be interpreted as the conventional quantum kernel method. (b) The distribution of outputs of the quantum circuit with $L = 10$ layers (Fig. 2) beyond the conventional quantum kernel.

V. CONCLUSION

We proposed a quantum tangent kernel (QTK) and a deep quantum tangent kernel which cannot be interpreted as the conventional quantum kernel methods described in Ref. [8]. The QTK is defined by applying the formulation of the NTK to parametrized quantum circuits. In Refs. [29,30], the equation describing the dynamics of training has been derived and solved by assuming there exists a limit where the parameters' change is small during training in the limit of a large number of qubits. In contrast, we found that in the limit of a large number of layers, the parameters of an overparametrized quantum circuit change only slightly from their initial values during training. This indicates that the output of an overparametrized quantum circuit can be linearly approximated, which validates the formulation of the QTK. By using this overparametrization, we can avoid the gradient descent for quantum circuits with a large number of parameters and easily optimize the parameters. Incidentally, the difference between the barren plateaus and the limit where parameters do not change much is discussed in Ref. [31]. Then, in order to increase the nonlinearity of a feature map, we introduced a multilayered data encoding that alternates between data encoding and a parametrized unitary. This encoding method increases nonlinearity of the feature map and improves the expressive power of kernels.

We demonstrated the performance of a shallow QTK and a deep QTK for a classification task. Using an ansatz-generated dataset generated by the quantum circuit for a deep QTK [Eq. (19)], we evaluate the performance by using a support vector machine with three kernels: a shallow QTK, a deep QTK, and a conventional quantum kernel. We showed that the SVM with the deep QTK outperforms the SVMs with

other kernels and the deep QTK has a feature map with high nonlinearity.

Our results imply that deep parametrized quantum circuits with repetitive data encoding unitary have a higher representation power and better performance for quantum machine learning than the conventional quantum kernel method. While we here employed an overparametrization limit and hence a deeper quantum circuit to take the neural tangent kernel approach, sophisticatedly trained neural networks, such as deep neural networks, provide a better performance in general than neural tangent kernels as known in the classical literature [32]. It gives us hope that the real fruit of quantum machine learning is not in the shallow or deep limit, but in the mild depth, which could be modeled by neither conventional nor tangent kernel methods. Therefore, better ansatz constructions and parameter optimization methods are crucially important.

Note added. Recently, Ref. [14] was published. At that time of writing this paper, the quantum circuits like Eq. (19) were not called a data re-uploading circuit. Therefore, we do not call a quantum circuit like Eq. (19) a data re-uploading circuit in this work.

ACKNOWLEDGMENTS

K.M. is supported by JST PRESTO Grant No. JPMJPR2019, JSPS KAKENHI Grant No. 20K22330, JST ASPIRE JPMJAP2319 and JSPS KAKENHI 24K16980. K.F. is supported by JST ERATO Grant No. JPMJER1601 and JST CREST Grant No. JPMJCR1673. This work is supported by MEXT Quantum Leap Flagship Program (MEXTQLEAP) Grants No. JPMXS0120319794 and JPMXS0118067394, and JST COINEXT Grant No. JPMJPF2014.

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[2] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[3] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).

[4] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature (London) **567**, 209 (2019).

[5] M. Schuld and N. Killoran, Quantum machine learning in feature Hilbert spaces, Phys. Rev. Lett. **122**, 040504 (2019).

[6] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Parameterized quantum circuits as machine learning models, Quantum Sci. Technol. **4**, 043001 (2019).

[7] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[8] M. Schuld, Supervised quantum machine learning models are kernel methods, arXiv:2101.11020.

[9] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, Power of data in quantum machine learning, Nat. Commun. **12**, 1 (2021).

[10] Y. Liu, S. Arunachalam, and K. Temme, A rigorous and robust quantum speed-up in supervised machine learning, Nat. Phys. **17**, 1013 (2021).

[11] T. Kusumoto, K. Mitarai, K. Fujii, M. Kitagawa, and M. Negoro, Experimental quantum kernel trick with nuclear spins in a solid, npj Quantum Inf. **7**, 94 (2021).

[12] K. Bartkiewicz, C. Gneiting, A. Černoch, K. Jiráková, K. Lemr, and F. Nori, Experimental kernel-based quantum machine learning in finite feature space, Sci. Rep. **10**, 12356 (2020).

[13] J. R. Glick, T. P. Gujarati, A. D. Corcoles, Y. Kim, A. Kandala, J. M. Gambetta, and K. Temme, Covariant quantum kernels for data with group structure, Nat. Phys. **20**, 479 (2024).

[14] S. Jerbi, L. J. Fiderer, H. P. Nautrup, J. M. Kübler, H. J. Briegel, and V. Dunjko, Quantum machine learning beyond kernel methods, Nat. Commun. **14**, 517 (2023).

[15] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, Nat. Phys. **15**, 1273 (2019).

[16] A. Pesah, M. Cerezo, S. Wang, T. Volkoff, A. T. Sornborger, and P. J. Coles, Absence of barren plateaus in quantum convolutional neural networks, Phys. Rev. X **11**, 041011 (2021).

[17] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, An initialization strategy for addressing barren plateaus in parametrized quantum circuits, Quantum **3**, 214 (2019).

[18] J. Kim, J. Kim, and D. Rosa, Universal effectiveness of high-depth circuits in variational eigenproblems, Phys. Rev. Res. **3**, 023203 (2021).

[19] M. Larocca, N. Ju, D. García-Martín, P. J. Coles, and M. Cerezo, Theory of overparametrization in quantum neural networks, Nat. Comput. Sci. **3**, 542 (2023).

[20] D. Wierichs, C. Gogolin, and M. Kastoryano, Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer, Phys. Rev. Res. **2**, 043246 (2020).

[21] A. Jacot, F. Gabriel, and C. Hongler, Neural tangent kernel: Convergence and generalization in neural networks, in *Proceedings of the 32nd Conference on Neural Information Processing Systems, Montreal, Canada* (NeurIPS, 2018), Vol. 31.

[22] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Phys. Rev. A **99**, 032331 (2019).

[23] J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington, Wide neural networks of any depth evolve as linear models under gradient descent, in *Proceedings of the 33rd Conference on Neural Information Processing Systems, Vancouver, Canada* (NeurIPS, 2019), Vol. 32.

[24] L. Deng, The MNIST database of handwritten digit images for machine learning research, IEEE Signal Process. Mag. **29**, 141 (2012).

[25] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nat. Commun. **9**, 4812 (2018).

[26] L. Banchi and G. E. Crooks, Measuring analytic gradients of general quantum evolution with the stochastic parameter shift rule, Quantum **5**, 386 (2021).

[27] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, General parameter-shift rules for quantum gradients, Quantum **6**, 677 (2022).

[28] S. Arora, S. S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang, On exact computation with an infinitely wide neural net, arXiv:1904.11955.

[29] J. Liu, K. Najafi, K. Sharma, F. Tacchino, L. Jiang, and A. Mezzacapo, Analytic theory for the dynamics of wide quantum neural networks, Phys. Rev. Lett. **130**, 150601 (2023).

[30] J. Liu, F. Tacchino, J. R. Glick, L. Jiang, and A. Mezzacapo, Representation learning via quantum neural tangent kernels, PRX Quantum **3**, 030323 (2022).

[31] J. Liu, Z. Lin, and L. Jiang, Laziness, barren plateau, and noise in machine learning, Mach. Learn.: Sci. Technol. **5**, 015058 (2024).

[32] Z. Li, R. Wang, D. Yu, S. S. Du, W. Hu, R. Salakhutdinov, and S. Arora, Enhanced convolutional neural tangent kernels, arXiv:1911.00809.